memory device that stores data. For example, the memory device **120** may be a flash memory device that includes a plurality of flash memory cells. Alternatively, the memory device **120** may be a memory device, which includes a plurality of resistive memory cells, such as including but not limited to a resistive random access memory (ReRAM), a magneto resistive random access memory (MRAM), a phase-change random access memory (PRAM), or the like. Hereinafter, in describing example embodiments, the memory device **120** is assumed as a flash memory device, which includes a plurality of NAND or NOR flash memory cells.

[0029] In response to a write, read, or trim request from the host **100**, the memory controller **110** may control the memory device **120** to write data in the memory device **120**, read data stored in the memory device **120**, or change valid data to invalid data, which is stored in the memory device **120**. For example, the memory controller **110** may supply a command CMD and a control logic CTRL to the memory device **120** to control a program (or write) operation, a read operation, an erase operation, and a trim operation, which are performed for the memory device **120**. Also, data DATA to be read and the read data DATA may be transmitted and received between the memory controller **110** and the memory device **120**.

[0030] To describe the trim operation, when an arbitrary file is erased from an upper level (i.e., a host side) by an application program of the host **100**, the arbitrary file may be treated as a file that has been erased by a file system of the host **100**. In an example embodiment, the host **100** may supply a trim request and address information of data, which is to be erased, to the memory controller **110** so as to erase data corresponding to the arbitrary file. In response to the trim request, the memory controller **110** may execute a trim command, generated for performing the trim operation, in background trim in a second priority after a priority of another command. In executing the trim command, the memory controller **110** may change data that corresponds to the arbitrary file to invalid data. The invalid data may be erased through garbage collection, which is to be performed later and further discussed in detail below.

[0031] Still referring to FIG. **1**, the memory controller **110** may include a flash transition layer (FTL) **115**. The FTL **115** may include system software (or firmware) that manages the write, read, erase, and trim operations of the flash memory device, and may be loaded into a working memory (not shown), which is included in the memory controller **110**. The firmware included in the FTL **115** may be driven by a processor (not shown) included in the memory controller **110**.

[0032] The FTL **115** may change a logical address to a physical address according to a data access request from the host **100**, and may supply the physical address to the memory device **120**. Also, the FTL **115** may perform a manage operation on various cell areas (for example, a chip, a block, a page, and/or the like) included in the memory cell array **120**, and for example, may perform garbage collection and a bad block managing operation on a plurality of blocks of the memory cell array (not shown) included in the memory device **120**.

[0033] Moreover, the HT **115** may include a command scheduler **115***a* and a workload calculator **115***b*. The memory controller **110** may receive a write request, a read request, and a trim request from the host **100** and may supply a plurality of commands, respectively corresponding to the received requests, to the memory device **120**. In an example embodiment, the command scheduler **115***a* may schedule the plurality of commands according to a command scheduling method in order for the plurality of commands to be efficiently executed. The command scheduling method may denote that a command is scheduled based on at least one of a priority of command executions and a quota of command execution. The quota of command execution may denote at least one of the amount of data, which is processed when one command is executed. And, the number of successive executions of the same command, which is continuously executed, may denote among a plurality of commands, The command scheduler **115***a* may change a command scheduling method, based on a workload level to be described below. That is, the command scheduler **115***a* may change at least one of a priority of command executions and a quota of command execution, which are a criterion for scheduling a command. Changing a quota of command execution may denote changing at least one of the amount of data, which is processed when one command is executed, and the number of successive executions of the same command, which is continuously executed, among a plurality of commands.

[0034] Still referring to FIG. **1**, the workload calculator **115***b* may calculate a workload level in an operation for performing garbage collection. The workload calculator **115***b* may calculate the workload level at a certain period as well as when performing garbage collection. In an example embodiment, the workload calculator **115***b* may calculate the valid page ratio of blocks for a victim block list or number of valid pages of blocks for a victim block list, which is generated for selecting at least one victim block for which garbage collection is to be performed. The workload calculator **115***b* may calculate the workload level based on the calculated valid page ratio or number, and may treat the calculated valid page ratio or number as the workload level. In another example embodiment, the workload calculator **115***b* may calculate the ratio or number of the non-free blocks instead of the free blocks among a plurality of blocks included in the memory device **120**. The workload calculator **115***b* may calculate the workload level based on the calculated ratio or number of the non-free blocks, and may treat the calculated ratio or number of the non-free blocks as the workload level.

[0035] The victim block list may be generated before the workload calculator **115***b* calculates the valid page ratio. And, the workload calculator **115***b* may calculate the valid page ratio of blocks for a victim block list. For example, in FIG. **5**A the valid page ratio of blocks for the victim block list is 6/12 because the total number of pages of blocks for the victim block list is 12, and the total number of valid pages of blocks for the victim block list is 6. As such, the valid page ratio and the invalid page ratio may be defined based on the number of pages of blocks for the victim block list.

[0036] However, this is merely an example embodiment, and the example embodiments are not limited thereto. In other example embodiments, the workload calculator **115***b* may calculate the workload level based on the invalid page ratio or number of blocks for the generated victim block list, and may calculate the workload level based on the ratio or number of free blocks among the plurality of blocks included in the memory device **120**. In an another example embodiment, the workload calculator **115***b* may calculate